Short-Term Load Forecasting for Industrial Customers Based on TCN-LightGBM

Yuanyuan Wang, *Member*, *IEEE*, Jun Chen, Xiaoqiao Chen, Xiangjun Zeng, *Senior Member*, *IEEE*, Yang Kong, Shanfeng Sun, Yongsheng Guo, Ying Liu

***Abstract*--Accurate and rapid load forecasting for industrial customers has been playing a crucial role in modern power systems. Due to the variability of industrial customers' activities, individual industrial loads are usually too volatile to forecast accurately. In this paper, a short-term load forecasting model for industrial customers based on the Temporal Convolutional Network (TCN) and Light Gradient Boosting Machine (LightGBM) is proposed. Firstly, a fixed-length sliding time window method is adopted to reconstruct the electrical features. Next, the TCN is utilized to extract the hidden information and long-term temporal relationships in the input features including electrical features, a meteorological feature and date features. Further, a state-of-the-art LightGBM capable of forecasting industrial customers' loads is adopted. The effectiveness of the proposed model is demonstrated by using datasets from different industries in China, Australia and Ireland. Multiple experiments and comparisons with existing models show that the proposed model provides accurate load forecasting results.**

***Index Terms*--Short-term load forecasting, industrial customers, temporal convolutional network, light gradient boosting machine.**

I. Introduction

THE forecasting of the power demand is of crucial importance for the development of modern power systems. Most countries will deploy or are deploying or have deployed the transition from a regulated operating scheme to a deregulated power market, and this transition requires shifting load forecasting from the supply-side to the demand-side. On the demand-side, industrial customers with huge impacts on the power grid consume an enormous amount of electricity energy.

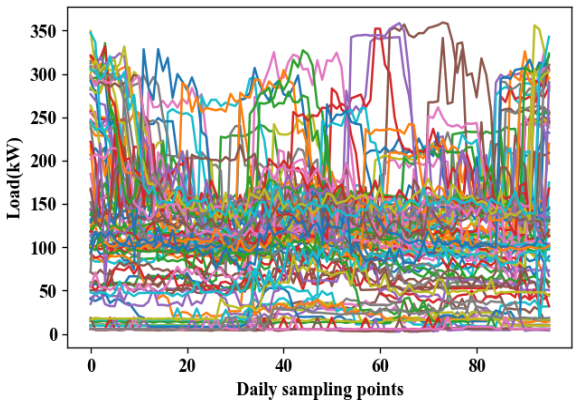


Fig. 1. Daily load profiles for a chemical industry customer in three months.

Taking China as an example, according to the statistics released in 2019 by the National Energy Administration of the People's Republic of China, the electricity consumption of industrial customers accounted for approximately 69% of the total electricity consumption of the whole society [1]. Under the huge demand for electricity, the peak-valley time-of-use tariff policy was formulated by China to achieve the peak shaving and valley filling of the electricity load. Since the electricity price during the peak period is much higher than that during the off-peak period [2], industrial customers have a strong desire to adjust their production planning in advance to reduce the electricity consumption and costs during the peak period. Accurate short-term load forecasting could help industrial customers forecast their future load variations, guide the industrial customers to adjust their production planning in advance, avoid the electricity peak and save the electricity costs. Therefore, the load service entities (LSEs) [3] are supposed to perform the short-term load forecasting to help industrial customers adjust their production plans.

However, from the technical aspect, there are great challenges in accurate load forecasting for industrial customers to adjust future production planning due to the continuous development of the industry and the increasing variability of customers' activities. Fig. 1. shows the daily load profiles for a chemical industry customer in three months. It is observable that the load profiles are volatile and irregular. In addition, the variability of industrial customers' activities and the volatility of industrial loads are also demonstrated in [4]. Therefore, how to forecast the loads of

This work is supported by the National Natural Science Foundation of China (No. 51777014). Hunan Provincial Key Research and Development Program (No. 2018GK2057). Research projects funded by Department of Education of Hunan Province of China under Grant (18A124). Changsha Science and Technology Project (kq1901104). Hunan Graduate Research and Innovation Project (CX20190686). Changsha University of Science and Technology Research and Innovation Project (CX2020SS52). *(Corresponding author: Xiaoqiao Chen.)*

Y. Y. Wang, J. Chen, X. J. Zeng, Y. Kong, S. F. Sun, Y. S. Guo, and Y. Liu are with the Hunan Province Key Laboratory of Smart Grids Operation and Control (School of Electrical and Information Engineering, Changsha University of Science and Technology), Changsha 410114, Hunan, P. R. China (e-mail: yuanyuan.wang.1980@ieee.org; 493322044@qq.com; zengxiangjun@ieee.org; 798266099@qq.com; 1254277142@qq.com; 1278065111@qq.com; 935086259@qq.com;).

X. Q. Chen is with Computing and Mathematical Science Department, California Institute of Technology (e-mail: xqchen@caltech.edu).

industrial customers is a key point for the Chinese electricity sector in order to meet the ambitious electricity market target.

The invention and application of various forecasting techniques promote the progress of short-term load forecasting. These models are mainly split into two categories. The first category is the time series models and the second category is the machine learning models. The time series models mainly include the exponential smoothing model [5], [6], the linear regression (LR) model [7], [8] and the autoregressive integrated moving average (ARIMA) model [9]-[11]. Although the time series models have simple structures and fast training, they cannot reflect the nonlinear characteristics of the load series. To solve the defects of the time series model, machine learning models have attracted more attention in load forecasting. Jiang *et al*. [12] propose a hybrid forecasting model based on the support vector regression (SVR) and hybrid parameter optimization algorithms. Their experiments demonstrate that the SVR model tuned by the two-step hybrid optimization algorithm achieves better performance than some classic models in short-term load forecasting. The Elman neural network (ELM) is used for short-term load forecasting in [13] and [14]. Raza *et al*. [14] propose an ensemble forecast framework (ENFF) with a systematic combination of an ELM, a feedforward neural network (FNN) and a radial basis function (RBF) neural network. The ensemble structure makes full use of multiple model advantages and improves the stability of short-term load forecasting. Kong *et al*. [15] introduce the long short-term memory neural network (LSTM) into short-term residential load forecasting. The advanced architecture of LSTM effectively solves the defect that the traditional recurrent neural network (RNN) is limited to learning short-term temporal correlations. The results indicate that the LSTM model outperforms the comparison models. A multi-layer bidirectional recurrent neural network model based on LSTM and gated recurrent unit (GRU) is proposed in [16] to predict short-term power load. The performance of the model is verified on two datasets. In real practice, industrial customers have numerous electrical equipment and large production scales, which results in rich electricity consumption data. However, the aforementioned models have difficulties effectively utilizing and extracting the feature information in the electricity consumption data.

Facing the problem, feature extraction techniques are considered to be feasible methods. A series of studies are dedicated to combining feature extraction techniques with prediction models for short-term load forecasting. Conventional models include the principal component analysis (PCA) [17], [18], the factor analysis (FA) [19] and the stationary wavelet transform (SWT) [20]. However, their extraction abilities need to be further improved since these models are unsuitable for extracting deep features. Recently, deep learning has been used extensively in feature extraction. Kang *et al.* [21] adopt a stacked auto-encoder (SAE) to extract the hidden features from the historical load data and predict the load for the working day with the GRU model. A hybrid model based on a convolutional neural network (CNN) and LSTM is introduced in [22] to forecast the short-term electric load consumption by individual households. Imani *et al.* [23] develop an LSTM-based feature extraction technology. The experimental results show that the features extracted by the individual LSTM model provide good forecasting results. Lv *et al.* [24] utilize the bi-directional gated recurrent unit (Bi-GRU) model, with the ability to simultaneously process past and future information, to extract the temporal and nonlinear features in the input data. In fact, the electricity consumption data of industrial customers is time-varying. The historical features over a long-time-range have an effect on the load at the current time. However, most techniques do not take the temporal correlations between input features over a long-time span into account, which makes the load forecasting model lose sufficient prior knowledge. Although the LSTM and Bi-GRU model can learn the long-term temporal correlations, the feature extraction capability still needs to be improved due to the lack of convolution.

Based on the above analysis, the latest temporal convolutional network (TCN) model is introduced. The model is used extensively in many fields such as pattern recognition [25], [26], anomaly detection [27] and mental assessment [28], but its application to the feature extraction task for load forecasting is relatively limited. Due to the integration of both the parallel feature processing of the CNN and the time-domain modeling capability of the RNN [29], the TCN is superior in extracting long-term time series features. In addition, a light gradient boosting machine (LightGBM) model [30] is selected to conduct load forecasting for industrial customers. The parallel mechanism of the data and features in the LightGBM model assists in the processing and forecasting of large-scale electricity consumption data. In this paper, an industrial customer load forecasting model based on the TCN and LightGBM is proposed. The influence factors including electrical features, a meteorological feature and date features are analyzed. Further, the actual variation and fluctuation trend of the electrical features are captured by a fixed-length sliding time window [31]. Next, the TCN model, which is able to extract the hidden information and temporal relationship in the features is utilized to effectively reduces redundant features and improves the load forecasting performance. Finally, the LightGBM model is introduced to forecast the load variation of industrial customers. Experimental results show that the hybrid model based on the TCN and LightGBM performs better on the load forecasting for industrial customers than the other listed contrast models. The main contributions of this paper are described as follows:

(1) We analyze multiple feature factors affecting industrial customer load forecasting in this paper. A fixed-length sliding time window is employed to capture the actual variation and fluctuation trend in the features.

(2) For the feature extraction task in load forecasting, a state-of-the-art TCN model with the integration of both the parallel feature processing and the time-domain modeling capability is proposed. The TCN model solves two problems

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

due to its special convolutional structure and residual block. First, it is able to extract the deep features and long-term temporal relationships. Second, it avoids the vanishing or exploding gradient during the deep network training.

(3) A pioneer study of applying LightGBM model into the short-term load forecasting for industrial customers is presented. Considering the amount of industrial customer data in practical application, LightGBM model is able to address the large-scale data prediction problem with the histogram algorithm and GOSS technology.

(4) A load forecasting model based on the TCN and LightGBM is proposed. This paper extends the application of the proposed model to multiple different types of industrial customers (medical industry, plastic products industry and coal mining industry) as well as a public dataset from the Irish smart meter project. Experimental results prove that the proposed model achieves better forecasting performance than other listed models in the short-term load forecasting task for industrial customers.

(5) This paper is an academic research based on the actual industrial demands. In real engineering practice, industrial customers in China need to adjust their production plans in advance to reduce the electricity consumption during peak periods. Only by accurately predicting the load can industrial customers make the best production plan. If the load forecasting has a large deviation, it will increase the electricity cost of industrial customers. The proposed model effectively solves practical industrial demands and brings important practical significance.

The remainder of this paper is organized as follows. In Section II, a short-term load forecasting framework for industrial customers based on the TCN-LightGBM model is introduced. Section III presents the experimental settings and comparative analysis of each model. The conclusion of this paper and the future work are given in Section IV.

II. Load Forecasting Framework for Industrial Customers Based on TCN-LightGBM

A. Temporal Convolutional Network

The temporal convolutional network (TCN) proposed by Bai *et al* in 2018 is an algorithm for processing time series data [32]. The causal convolution, dilated convolution and residual block are introduced in the TCN, which solve the problem of extracting long-term time series information. Each structure is described in detail as follows.

1) Causal Convolutions

The causal convolution is the key architecture of the TCN. Fig. 2 shows the structure of the causal convolution stack. For one-dimensional time series input $X = (x_0, x_1, ..., x_t, \ldots, x_T)$, the output y_t at time t only depends on the inputs from the current time x_t and the partial past time (i.e., x_{t-1}, x_{t-2}, x_{t-3}), while not on any future inputs (i.e., x_{t+1}, x_{t+2}, x_{t+3}, …, x_T). On the one hand, the output information of the network only impacted by past input information, avoiding the "leakage" from future to past [32]. On the other hand, the causal convolution is easily limited by the receptive field, which means that the output can only receive information from a short history sizes to make a prediction.

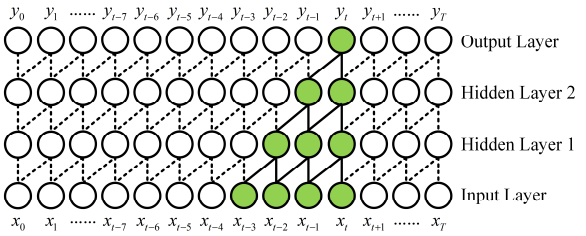


Fig. 2. The structure of the causal convolution stack.

2) Dilated Convolutions

To solve the limited receptive field problem, the TCN introduces the dilated convolution on the basis of causal convolution [33]. For a one-dimensional time series input $X = (x_0, x_1, ..., x_t, \ldots, x_T)$ and a filter $f:\{0, 1, 2,..., n\text{-}1\}$, the dilated convolution operation $H(\cdot)$ of the sequence element T is defined as follows:

$$H(T) = (X *_d f)(T) = \sum_{i=0}^{n-1} f(i) \cdot x_{T-d \cdot i} \tag{1}$$

where n denotes the filter size, d represents the dilation factor and $T\text{-}d\cdot i$ accounts for the direction of the past.

By increasing the filter size n and dilated factor d, the TCN can effectively expand the receptive field, which enables an output at the top layer to receive a wider range of input information. In addition, by processing the same filter in each layer in parallel, the computational efficiency of the whole model can also be improved. Fig. 3 presents the structure of the dilated causal convolution stack with filter size $n = 2$ and dilation factor $d = [1, 2, 4]$. After the dilated convolution is added, the output y_t at time t can receive the information of inputs x_{t-7}, x_{t-6}, ..., x_t.

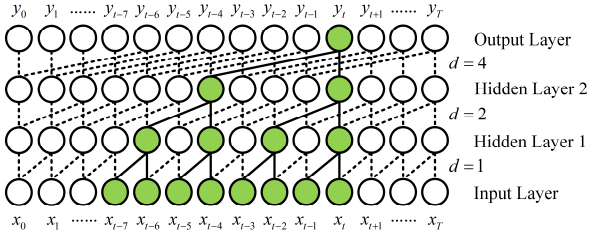


Fig. 3. The structure of the dilated causal convolution stack.

3) Residual Blocks

In addition to adjusting the filter size n and dilation factor d, the receptive field size of TCN can also be expanded by increasing the number of hidden layers. However, very deep networks will affect the stability of model training and occur the vanishing gradients. To address this issue, the residual block is adopted by TCN [34]. The details of the residual block are shown in Fig. 4(a).

One branch of the residual block performs a transformation operation $F(\cdot)$ on the input $X^{(h-1)}$, adding a branch to perform a simple $1\times1Conv$ transformation to maintain the number of feature maps consistency in parallel with the existing branch. The output $X^{(h)}$ of the h-th residual block can be expressed as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

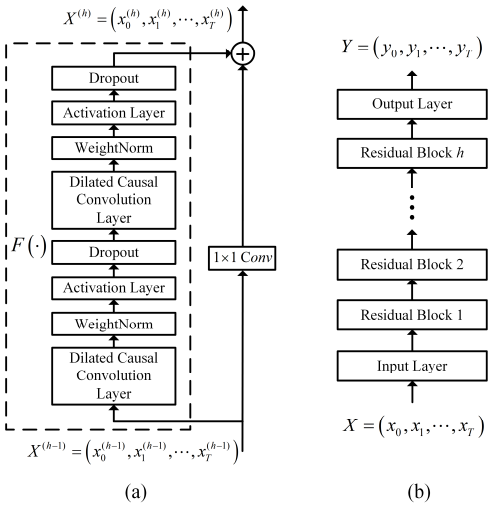


Fig. 4. The internal structure of the residual block and deep TCN. (a) Residual block. (b) A deep TCN.

$$X^{(h)}=\delta\left(F\left(X^{(h-1)}\right)+X^{(h-1)}\right) \tag{2}$$

where $\delta(\cdot)$ represents an activation operation. $F(\cdot)$ is a series of transformation operations whose structure includes the dilated causal convolution layer, the WeightNorm, the activation layer and dropout. Specifically, the dilated causal convolution layer is composed of the aforementioned causal convolution and dilated convolution, which is used to extract the hidden features from the input. The WeightNorm is used to improve the training speed by limiting the range of weights. A rectified linear unit (ReLU) [35] with good convergence is adopted by the activation layer. Dropout is used for regularization to solve the over-fitting of the deep network.

Fig. 4(b) shows a deep TCN formed by stacking h residual blocks. Building a deep TCN enables the networks to look very far into the past to extract features, i.e., each convolution of the output layer can receive more information from the convolution of the input layer.

B. TCN Based Feature Extraction Network

To extract the long-time-range time series features suitable for load forecasting, Fig. 5 presents a feature extraction network based on the TCN illustrated in Fig. 4(b).

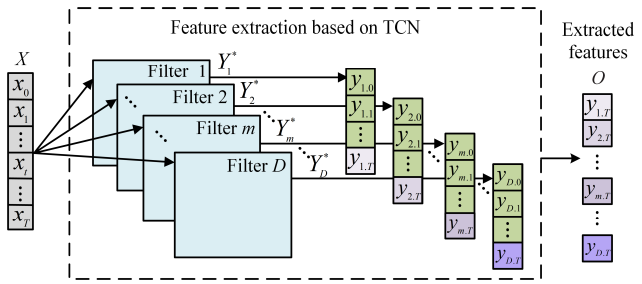


Fig. 5. An illustration of the feature extraction network based on the TCN.

The specific process of the network is as follows. First, the one-dimensional features $X = (x_0, x_1,..., x_t,..., x_T)$ is fed into D filters of TCN to obtain Y^* sized D-by-T, i.e., Y_1^*, Y_2^*,..., Y_m^*,..., Y_D^*, where the m-th vector is represented as $Y_m^* = (y_{m.0}, y_{m.1},..., y_{m.T})$, and m=1,2,...,D. Then, the last element of each one-dimensional vector Y_m^* is concatenated and fed to the fully connected layer. Finally, the extracted features $O = (y_{1.T}, y_{2.T}, ..., y_{m.T}, ..., y_{D.T})$ in the fully connected layer are used as the input of the LightGBM model (Section II-C) to forecast the industrial customers' loads. Notably, each eigenvector integrates only the elements that are most relevant to the task, and so the hidden information and long-term temporal relationship in the input features are extracted by the TCN model.

C. LightGBM Model

The LightGBM is a gradient boosting framework based on a decision tree algorithm proposed by Microsoft research in 2017 [30], which is widely employed in classification or regression tasks [36], [37]. In this paper, we adopted Gradient-based One-Side Sampling (GOSS) to narrow the search area for split points, a histogram-based algorithm to find the best split points and a leaf-wise growth strategy with depth limitation, thereby solving the problems of high computational complexity and large memory consumption in the gradient boosting decision tree (GBDT) model.

Conventional implementations of GBDT needs to, for every feature, scan the whole load data to compute the gain for each possible split point, which makes it unable to handle large amounts of data due to the huge time/memory consumption. To tackle this issue, GOSS technology narrows the split point search area by reducing the number of data instances or the number of features. In the technique, all data are sorted in descending order according to the absolute value of the gradient. Then, the top a×100% data with the largest gradients are extracted as subset P and b×100% data are randomly extracted from the rest of the data as subset Q. Therefore, the gain can be computed from a narrow area (i.e., the subset $P \cup Q$) extracted by GOSS.

The variance gain $G_j(v)$ of splitting feature j at point v is defined as follows:

$$\begin{cases} G_j(v)=\dfrac{1}{S}\left(G_{j1}(v)+G_{j2}(v)\right) \\ G_{j1}(v)=\dfrac{1}{S_l(v)}\left(\sum_{X_k\in P_l} g_k+\dfrac{1-a}{b}\sum_{X_k\in Q_l} g_k\right)^2 \\ G_{j2}(v)=\dfrac{1}{S_h(v)}\left(\sum_{X_k\in P_h} g_k+\dfrac{1-a}{b}\sum_{X_k\in Q_h} g_k\right)^2 \end{cases} \tag{3}$$

where S is the number of instances in the subset $P \cup Q$, $S_l(v)$ and $S_h(v)$ denote the number of instances whose value of feature j is less than or greater than v, respectively. $P_l=\{X_k \in P: X_{kj} \le v\}$, $Q_l=\{X_k \in Q: X_{kj} \le v\}$, $P_h=\{X_k \in P: X_{kj} > v\}$, $Q_h=\{X_k \in Q: X_{kj} > v\}$, and g_k is the negative gradient of instance X_k.

According to the definition of the gain, instances with larger gradients (i.e., under-trained instances) play a greater role in the computation of the gain. Compared with the conventional GBDT, the gain computing method described in (3) allows the model to pay more attention to the under-trained data while dropping those data with small gradients.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

With the same sampling rate, the GBDT algorithm with GOSS can lead to a more accurate gain estimation, thereby improving the load forecasting performance [30].

To find the optimal splitting point of the decision tree and decrease the time/memory consumption, the LightGBM adopts a more efficient histogram algorithm. First, the one-dimensional feature is divided into multiple regions and each region is formed as a bin, as shown in Fig. 6. Then, the multiple bins obtained are formed into a histogram. Each bin in the histogram stores two types of information, namely, the number of instances and the sum of the gradients. For datasets with multi-dimensional features, the LightGBM find the optimal splitting point of the node to be divided by scanning several histograms.

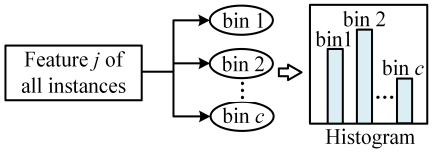


Fig. 6. The construction process of the histogram.

The traditional level-wise growth strategy for decision tree is shown in Fig. 7(a), which achieves tree growth by splitting the leaf nodes of each layer. However, the growth strategy will consume many computing resources to split the nodes with low information gain. The LightGBM adopts a leaf-wise growth strategy with depth limitation. As shown in Fig. 7(b), the strategy only splits the leaf nodes with the largest gain at each time. This strategy not only reduces the number of node splits but also avoids the unnecessary memory and computation consumption caused by the level-wise strategy. Moreover, the leaf-wise growth strategy imposes a depth limitation on the decision tree in order to prevent the over-fitting problem caused by the extremely deep decision tree.

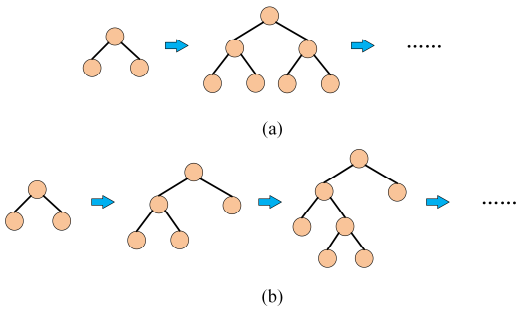


Fig. 7. Two growth strategies for decision trees. (a) Level-wise growth strategy. (b) Leaf-wise growth strategy.

D. The Load Forecasting Framework Based on TCN-LightGBM Model

On the one hand, the load data of most industrial customers has potential temporal correlations [4]. The TCN can extract the temporal correlations in the features due to the integration of the CNN's extraction ability and the RNN's time-domain modeling ability [29]. Moreover, the LightGBM is able to handle large-scale data accurately and quickly considering the load data scale of industrial customers in real practice. Therefore, the proposed model can make predictions for industrial customers based on the extracted feature information. The total research framework based on the TCN-LightGBM model is illustrated in Fig. 8. The framework is established by five steps, including missing data processing, feature selection, feature preprocessing, TCN-based feature extraction and LightGBM-based load forecasting. Each step in the framework is described in detail as follows:

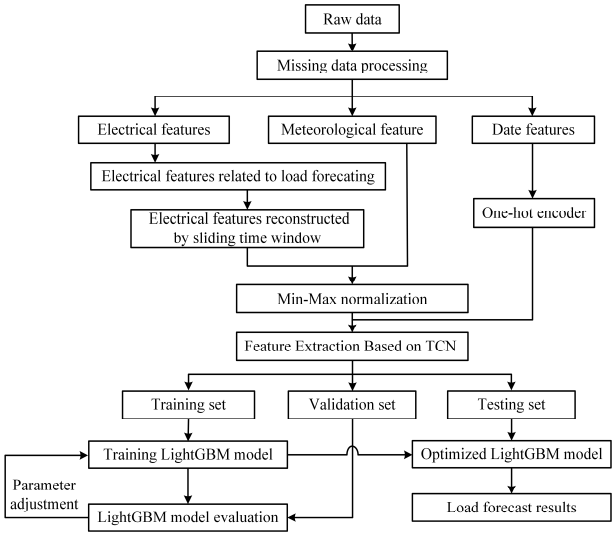


Fig. 8. The load forecasting framework based on TCN-LightGBM model.

1) Missing Data Processing

The raw data usually have a large amount of missing data because of interrupted signal transmissions or acquisition equipment failures. To avoid the adverse impact of missing data on load forecasting, the following methods is adopted: if the instance has a large proportion of missing data, the instance is deleted. Instead, missing data are filled with data from the same moment of the previous day.

2) Feature Selection

In this paper, we consider six main electrical features, namely, the load, current, active power, reactive power, power factor and voltage, because these features may have great impacts on the loads. To study the influence of different electrical features on the load and select the principal features, the Pearson correlation coefficient [38] is chosen to calculate the correlation between the electrical features and load, as shown in Table I. It is obvious that the "load" electrical feature is correlated with the current, active power and reactive power with correlation coefficient scores over 0.9, and it can be selected as the electrical features related to load forecasting. However, the power factor and voltage features are uncorrelated with the load, with correlation coefficient scores no more than 0.6, and thus they can be deleted. Further, Table II analyzes the forecasting results of the proposed model with or without the power factor and voltage. The experimental Settings are described in section III-B. It is also proved that the electrical features selected via the Pearson correlation coefficient are able to bring better forecasting performance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

TABLE I
PEARSON CORRELATION COEFFICIENT BETWEEN ELECTRICAL FEATURES AND THE "LOAD"

Electrical Feature	Correlation Coefficient
Current	0.999
Active power	0.984
Reactive power	0.906
Power factor	0.600
Voltage	0.335

TABLE II
FORECASTING RESULTS OF PROPOSED MODEL WITH DIFFERENT FEATURES

Features	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)	Time (s)
All electrical features	12.69	6.42	21.85	59.10
Electrical features selected via Pearson correlation coefficient	12.57	6.43	21.67	37.33

In addition to the electrical features, the meteorological factor is also considered as an associated factor influencing the load variation of industrial customers. Fig. 9 shows the load and temperature profiles for an industrial customer in 2018. It is obvious that the load of the industrial customer is positively correlated with the temperature change. Similarly, the Pearson correlation coefficient also proves the correlation between the temperature and load. The correlation coefficient score reaches 0.831. Therefore, the meteorological feature is chosen as important characteristics of load forecasting in this paper.

In addition, the date factors referring to the days of the week, month and holidays are considered. Fig. 10 illustrates the daily load profile for an industrial customer on February 8-22, 2018. It is apparent that the load reaches a lower level due to the production planning reduction during the Spring Festival (from 15-Feb-2018 to 21-Feb-2018). Therefore, it is appropriate to regard the date factors as relevant factor of industrial customers load forecasting.

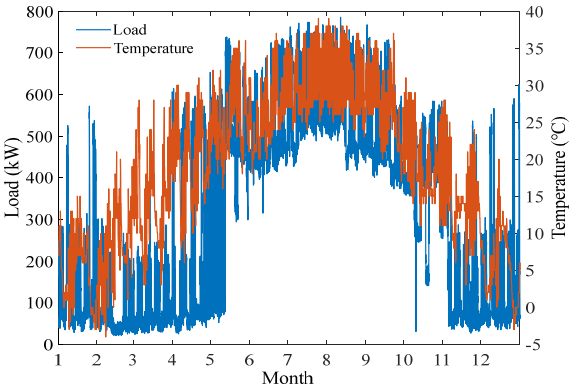


Fig. 9. Load and temperature profiles for an industrial customer in 2018.

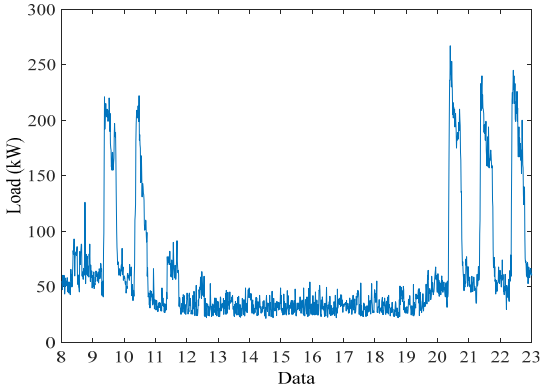


Fig. 10. Daily load profile for an industrial customer on February 8-22.

3) Feature Preprocessing

In this paper, the features are split into electrical features, a meteorological feature and date features. Each type of feature is processed according to the method described in Table III.

TABLE III
RAW INPUT DATA AND PROCESSING METHODS

Type of feature	Feature	Symbol	Processing method
Electrical features	Load	P_L	Reconstructing electrical features using the sliding time window method, and they are normalized by the Min-Max method
	Current	I_O	
	Active power	P_{AT}	
	Reactive power	P_{RT}	
Meteorological feature	Temperature	T_e	Cubic spline interpolation is used and the data then are normalized by the Min-Max method
Date features	Week	W	Feature mapping: 1-7 represent Monday to Sunday
	Month	R	Feature mapping: 1-12 represent January to December
	Holiday	H	Feature mapping: 0 for a non-holiday and 1 for a holiday

Among all types of features, the electrical features have an obvious variation in a short time period with the operating mode adjustment. Therefore, the electrical features in the previous time may have a significant impact on load forecasting at the current time. In this paper, we introduce a fixed-length sliding time window [31] that shares raw input data with their artificial constructed series average values, thus enabling the networks to look very far into the past to extract time-varying features. Specifically, the reconstructed electrical features M is composed of the smart meter data M_1 (96 instances per day) and the average values M_2 (96 artificial constructed instances per day) of the electrical features in the sliding window, which respectively reflect the actual variation and fluctuation trend of the electrical features, as shown in (4)-(7). By doing so, the electrical features can be deeply captured to improve the load forecasting performance.

$$M = [M_1, M_2] \tag{4}$$

$$\begin{cases} M_1 = [X_1', X_2', \cdots, X_{96}'] \\ M_2 = [\bar{X}_1', \bar{X}_2', \cdots, \bar{X}_{96}'] \end{cases} \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

$$\begin{cases} X'_1 = \left[X'_{1(w-96)}, X'_{2(w-96)}, X'_{3(w-96)}, X'_{4(w-96)} \right] \\ X'_2 = \left[X'_{1(w-95)}, X'_{2(w-95)}, X'_{3(w-95)}, X'_{4(w-95)} \right] \\ \cdots\cdots \\ X'_{96} = \left[X'_{1(w-1)}, X'_{2(w-1)}, X'_{3(w-1)}, X'_{4(w-1)} \right] \end{cases} \tag{6}$$

$$\begin{cases} \overline{X}'_1 = \left[\dfrac{X'_{1(w-1)} + \cdots + X'_{1(w-96)}}{96}, \cdots, \dfrac{X'_{4(w-1)} + \cdots + X'_{4(w-96)}}{96} \right] \\ \overline{X}'_2 = \left[\dfrac{X'_{1(w-1)} + \cdots + X'_{1(w-95)}}{95}, \cdots, \dfrac{X'_{4(w-1)} + \cdots + X'_{4(w-95)}}{95} \right] \\ \cdots\cdots \\ \overline{X}'_{96} = \left[X'_{1(w-1)}, X'_{2(w-1)}, X'_{3(w-1)}, X'_{4(w-1)} \right] \end{cases} \tag{7}$$

The structure of sliding time window is shown in Fig. 11. If we want to forecast the load $X'_{4(w)}$ corresponding to the w-th instance, the 192×4 electrical features in the sliding window SW_w (including M_1 and M_2) are reconstructed as M. Then, the sliding window SW_w shifts one step to the right to reconstruct SW_{w+1}, with which the load $X'_{4(w+1)}$ corresponding to the $(w+1)$-th instance can be predicted. The step size of the sliding window is set to 1, which means that the sliding window in Fig. 11 slides to the right one step at a time. This process is repeated until all the loads are predicted.

Since the TCN model is sensitive to the data scale, continuous features including the reconstructed electrical features and meteorological feature are normalized by the Min-Max method [39], while discrete features such as date features are encoded by the One-hot encoder.

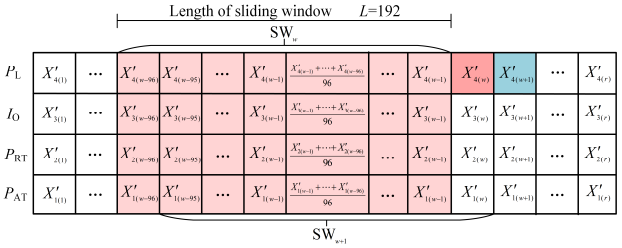


Fig. 11. The structure of sliding time window.

4) TCN Based Feature Extraction

The input data contains rich features X (including 768-dimensional electrical features rebuilt by a sliding time window, a 1-dimensional meteorological feature and 3-dimensional date features), which pose challenges to the load forecasting for industrial customers. Fortunately, the TCN-based feature extraction network described in Fig. 5 can effectively extract the features X of the input data. It not only extracts the hidden information and long-term temporal relationship within the features but also reduces the feature dimensions of the input data.

$$X = [R, W, H, T_c, M] \tag{8}$$

5) LightGBM Based Load Forecasting

The data obtained from the feature extraction are divided into a training set, verification set and testing set. The training set is used to train the LightGBM model and the validation set is used for the model parameter tuning. After repeated iterations, the testing set is input into the optimized LightGBM to forecast industrial customers load and evaluate the model performance.

E. Performance Evaluation

In this paper, the Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are used to evaluate the performance of the proposed model. The statistical metrics are defined as follows:

$$\gamma_{rmse} = \sqrt{\frac{1}{N}\sum_{w=1}^{N}\left(Z_w - \hat{Z}_w\right)^2} \tag{9}$$

$$\gamma_{mae} = \frac{1}{N}\sum_{w=1}^{N}\left|Z_w - \hat{Z}_w\right| \tag{10}$$

$$\gamma_{mape} = \frac{1}{N}\sum_{w=1}^{N}\left|\frac{Z_w - \hat{Z}_w}{Z_w}\right| \times 100\% \tag{11}$$

where N denotes the number of validation or testing instances. Z_w and $\hat{Z}_w$ represents the actual load and forecasted load of the w-th instance, respectively.

Each statistical metric has different pros and cons. The RMSE measures the accuracy by comparing the deviation between the forecasted and actual load. The metric maintains a uniform dimension with the load, but it is susceptible to outliers because of the sensitivity to larger or smaller errors. The MAE represents the average absolute error between the forecasted and actual load. Compared with the RMSE, the metric shows better robustness to outliers but the degree of prediction deviation cannot be fully reflected. The MAPE expresses the prediction accuracy by calculating the absolute error percentage. The metric considers the relative gap between the forecasted and actual load, but it is not applicable when the actual load is zero. Therefore, it is necessary to adopt multiple statistical metrics to evaluate the forecasting performance.

III. Case Study

A. Experimental Settings

Real-world smart meter data from different types of industrial customers (medical industry, plastic products industry and coal mining industry) with a temporal resolution of 15 minutes interval are used in this paper. The datasets for the medical and plastic products industry are collected in Hunan Province, China. For the coal mining industry, the dataset is obtained from New South Wales, Australia. In addition, a public dataset acquired from the Irish Smart Metering Electricity Customer Behaviour Trials (CBTs) [40] with a temporal resolution of 30 minutes interval is also used, which records the electricity demand of residential customers and small-to-medium industrial sites from 14-Jul-2009 to 31-Dec-2010 [41]. The temperature data can be obtained from the National Oceanic and Atmospheric Administration (NOAA) website. The temporal resolution of the temperature data is 1-hour interval and 24 instances are sampled every day. To maintain the temporal resolution consistent with the load time series, the temperature data are processed using

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

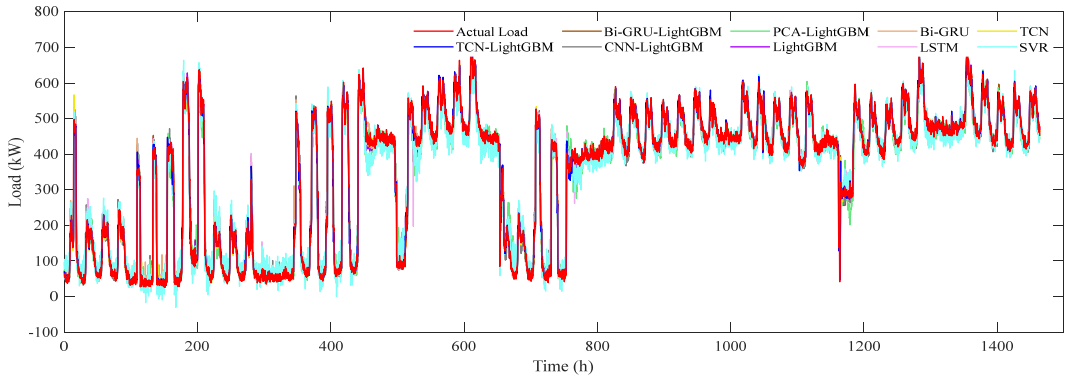


Fig. 12. Load forecasting profiles of all models during 01-Apr-2019 and 31-May-2019.

cubic spline interpolation. Finally, the datasets are split into a training set, a validation set and a testing set according to the proportion of 8:1:1, i.e., the training set accounts for 80%, the validation set accounts for 10% and the testing set accounts for 10%.

The proposed model and other listed contrast models are studied in this paper. Specifically, these contrast models include a statistical model (SVR), deep learning models (TCN, LSTM and Bi-GRU), a tree model (LightGBM) and hybrid models (PCA-LightGBM, CNN-LightGBM and Bi-GRU-LightGBM). Different parameter tuning strategies are adopted for different models. Since the SVR, LightGBM and PCA have fewer parameters, the Grid Search (GS) algorithm is used for their tuning. The parameters of deep learning models are tuned by previous experience considering that these models require more parameter tuning skills. All experimental models run in the Python 3.6 programming environment. The hardware is a PC with an Intel core i7-9700k CPU and 16GB of memory.

B. Experiment 1: Load forecasting for the medical industry customer

In this experiment, the medical industry data collected from 01-Nov-2017 to 31-May-2019 are employed to run simulations. The period spans over 577 days. Based on the splitting rules discussed above, the time range of the training set is from 01-Nov-2017 to 31-Jan-2019, the next two months from 01-Feb-2019 to 31-Mar-2019 are as the validation set and the last two months from 01-Apr-2019 to 31-May-2019 are as the testing set. All models are trained with the training set and optimized with the validation set. Finally, the parameters of each model are summarized as follows.

(1) TCN: The algorithm is built using the Keras library. The number of filters is 128, the size of the filter is 2, and the dilation factor is set to [1, 2, 4, 8, 16].

(2) LightGBM: The number of trees is 800, the maximum depth is 6, the number of leaves is 40, the learning rate is set to 0.025, the bagging fraction is 0.46, the feature fraction is 0.5, the chosen model optimizer is Adam and the boosting method is gbdt.

(3) CNN: The number of convolutional layers is 1, the number of filters in the convolutional layer is 16, the size of the convolutional kernel is 2, the number of fully connected layers is 3 and the number of neurons in the fully connected layer is set to 100/5/1.

(4) LSTM: The number of hidden layers is 6 and the number of hidden nodes is set to 100/100/100/50/50/50.

(5) Bi-GRU: The number of hidden layers is 4 and the number of hidden nodes is set to 100/100/50/50.

(6) SVR: The kernel function is linear and the penalty factor is set to 1.

(7) PCA: The algorithm is built by the Scikit-learn library and the number of principal components is 27.

The forecasting results of all models on the testing set are presented in Fig. 12. It is indicated that the industrial customers' loads are volatile, i.e., the electricity consumption during April (from 0 hours to 720 hours) and May (from 720 hours to 1465 hours) is completely different, which makes it challenging for the accurate forecasting of each model. To better evaluate the performance of the proposed model, the load of May 7 (from 1296 hours to 1392 hours) is selected for further analysis, as shown in Fig. 13. It shows that all models can roughly fit the actual load in the phase of rising or falling. At the peak, valley, stable or fluctuation range of the actual load, each model has different deviation, but the proposed model is able to fit and catch the trend of actual load.

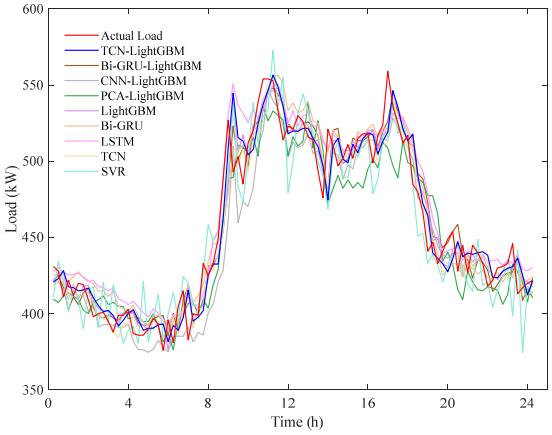


Fig. 13. Load forecasting profiles of all models on 7-May-2019.

The evaluation results of all models on the testing set are presented in Table IV. It can be observed that the LightGBM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

model is better than other individual models. The performance of the SVR model is relatively poor. In the hybrid models, the extraction capacity of the TCN is superior to PCA, the CNN and Bi-GRU according to the given statistical metrics. Among all the contrast models, the proposed model obtains the best prediction performance due to the excellent extraction capability of the TCN.

The computational complexity of each model on the testing set is provided in Table IV. It shows that the LightGBM model consumes the least time for load forecasting and the proposed model takes less time than the TCN model. Although the computational time of the proposed model is relatively higher, it is acceptable in practical application with the popularization of cloud computing.

TABLE IV
LOAD FORECASTING EVALUATION ON THE TESTING SET

Time	From 01-Apr-2019 to 31-May-2019			
Statistical metrics	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)	Computational time (s)
SVR	24.58	15.41	32.93	2.16
TCN	13.16	7.08	22.29	40.09
LSTM	16.23	8.59	26.38	0.47
Bi-GRU	16.59	9.43	25.68	0.93
LightGBM	12.93	6.60	21.73	0.24
PCA-LightGBM	25.08	13.51	37.92	1.79
CNN-LightGBM	17.24	9.62	26.53	0.49
Bi-GRU-LightGBM	13.37	6.84	22.24	6.36
Proposed model	**12.57**	**6.43**	**21.67**	37.33

In addition, the *t*-test [42] and Friedman test [43], [44] are implemented to verify the significance level of the proposed model. One-tail tests are performed at the significance level of α=0.05. All the test results are shown in Table V. The test results conclude that the proposed model is significantly different from the contrast models.

TABLE V
SIGNIFICANCE TESTS ON THE TESTING SET

Models	*t*-Test	Friedman Test
Proposed model vs. SVR	0.000	
Proposed model vs. TCN	0.000	
Proposed model vs. LSTM	0.000	
Proposed model vs. Bi-GRU	0.000	p=0.000
Proposed model vs. LightGBM	0.000	(Reject H_0)
Proposed model vs. PCA-LightGBM	0.000	
Proposed model vs. CNN-LightGBM	0.000	
Proposed model vs. Bi-GRU-LightGBM	0.001	

H_0: There is no significant difference between the models.

Considering the huge differences in the load variation between different months, the experiment conducts comparative analysis and evaluate the load forecasting metrics in two special months (April and May) to further verify the forecasting performance. Table VI shows the load forecasting results for two representative days in April and May. It is obvious that the proposed model achieves the best forecasting results in respect of all the statistical metrics. In the individual models, the TCN model is slightly superior to the LightGBM model.

TABLE VI
LOAD FORECASTING EVALUATION IN TWO REPRESENTATIVE DAYS

Time	One day in April 2019			One day in May 2019		
Statistical metrics	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)
SVR	32.45	8.28	39.53	25.19	6.30	31.88
TCN	9.61	2.17	11.87	13.75	3.43	20.58
LSTM	13.04	3.03	16.40	15.79	3.97	20.74
Bi-GRU	18.37	4.13	22.43	17.64	4.31	23.67
LightGBM	10.23	2.35	12.69	14.35	3.62	21.53
PCA-LightGBM	13.37	3.06	17.97	42.53	12.56	54.21
CNN-LightGBM	9.16	2.09	12.11	14.79	3.70	20.34
Bi-GRU-LightGBM	9.51	2.13	11.92	14.26	3.47	20.61
Proposed model	**8.59**	**1.93**	**10.61**	**13.20**	**3.24**	**19.01**

Additionally, the computational complexity of each model in two representative days is given in Table VII. It can be seen that the proposed model requires more time consumption, but it is acceptable considering the improvement in the forecasting accuracy. Further, the *t*-test and Friedman test are employed to evaluate the significance of the model, as shown in Tables VIII and IX. One-tail tests are performed at the significance level of α=0.05. The results in Tables VIII and IX also indicate that the proposed model is significantly different from the contrast models at the significance level of α=0.05.

TABLE VII
COMPUTATIONAL COMPLEXITY FOR EACH MODEL IN TWO REPRESENTATIVE DAYS

Models	Computational time (s)
SVR	0.08
TCN	1.90
LSTM	0.02
Bi-GRU	0.04
LightGBM	0.02
PCA-LightGBM	0.26
CNN-LightGBM	0.06
Bi-GRU-LightGBM	0.80
Proposed model	1.80

TABLE VIII
SIGNIFICANCE TESTS FOR ONE DAY IN APRIL 2019

Models	*t*-Test	Friedman Test
Proposed model vs. SVR	0.000	
Proposed model vs. TCN	0.000	
Proposed model vs. LSTM	0.002	
Proposed model vs. Bi-GRU	0.003	p=0.000
Proposed model vs. LightGBM	0.000	(Reject H_0)
Proposed model vs. PCA-LightGBM	0.000	
Proposed model vs. CNN-LightGBM	0.000	
Proposed model vs. Bi-GRU-LightGBM	0.028	

H_0: There is no significant difference between the models.

TABLE IX
SIGNIFICANCE TESTS FOR ONE DAY IN MAY 2019

Models	*t*-Test	Friedman Test
Proposed model vs. SVR	0.000	
Proposed model vs. TCN	0.010	
Proposed model vs. LSTM	0.002	
Proposed model vs. Bi-GRU	0.000	p=0.000
Proposed model vs. LightGBM	0.001	(Reject H_0)
Proposed model vs. PCA-LightGBM	0.000	
Proposed model vs. CNN-LightGBM	0.003	
Proposed model vs. Bi-GRU-LightGBM	0.001	

H_0: There is no significant difference between the models.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2020.3028133, IEEE Transactions on Power Systems

C. Experiment 2: Load forecasting for the plastic products industry customer

In this experiment, the plastic products industry data collected from 01-Feb-2018 to 31-May-2018 are utilized to run simulations. The period spans over 120 days. According to the aforementioned splitting rules, the training set is from 01-Feb-2018 to 17-Apr-2018, the validation set is from 18-Apr-2018 to 09-May-2018 and the testing set is from 10-May-2018 to 31-May-2018. After parameter tuning, the parameters of each model are summarized as follow.

(1) TCN: The algorithm is built using the Keras library. The number of filters is 64, the size of the filter is 2 and the dilation factor is set to [1, 2, 4, 8].

(2) LightGBM: The number of trees is 650, the maximum depth is 7, the number of leaves is 90, the learning rate is set to 0.04, the bagging fraction is 0.4, the feature fraction is 0.9, Adam is chosen as the model optimizer and the boosting model is the gbdt.

(3) CNN: The number of convolutional layers is 1, the number of filters in the convolutional layer is 8, the size of the convolutional kernel is 2, the number of fully connected layer is 2 and the number of neurons in the fully connected layer is set to 32/1.

(4) LSTM: The number of hidden layers is 1 and the number of hidden nodes is set to 30.

(5) Bi-GRU: The number of hidden layers is 1 and the number of hidden nodes is set to 30.

(6) SVR: The kernel function is linear and the penalty factor is set to 0.01.

(7) PCA: There are 113 principal components.

The statistical metrics of each model on the testing set are shown in Table X. Compared with experiment 1, the prediction effect of the proposed model is more significant and competitive. Furthermore, Fig. 14 shows a comparison between the forecasting and actual load of each model on 18-May-2018. It is observed that the load forecasting profile of the proposed model is able to match the actual load. Especially, the occasional fluctuation can be captured by the proposed model. However, other contrast models show different deviation in response to sudden variations or spikes in the profiles.

For the plastic products industry data, the computational time of each model is also calculated, as shown in Table X. Because of the less testing set data, the computational times are significantly reduced compared to Table IV. The *t*-test and Friedman test are adopted to demonstrate the significance of the proposed model. One-tail tests are performed at the significance level of α=0.05. Two test results are shown in Table XI. It can be found that the proposed model receives the significant differences from the contrast models in the plastic products industry data, i.e. the hypothesis H_0 is rejected.

D. Experiment 3: Load forecasting for the coal mining industry customer

The coal mining industry data collected from 01-May-2010 to 31-Jul-2010 is adopted to run simulations. The

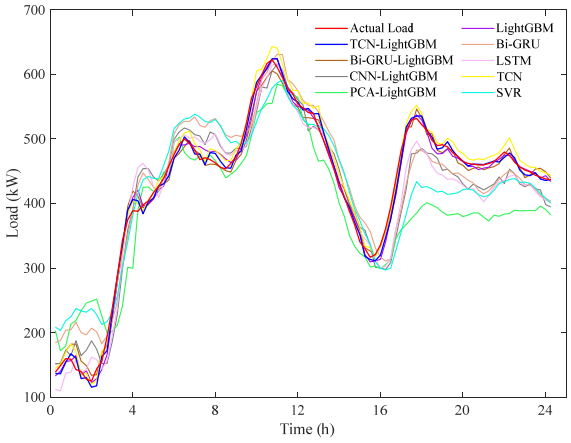


Fig. 14. Load forecasting profiles of all models on 18-May-2018.

TABLE X
LOAD FORECASTING EVALUATION ON THE TESTING SET

Time	From 10-May-2018 to 31-May-2018			
Statistical metrics	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)	Computational time (s)
SVR	28.81	9.08	38.7	0.17
TCN	12.93	3.57	17.25	4.89
LSTM	26.32	6.53	34.07	0.19
Bi-GRU	23.99	7.74	32.99	0.11
LightGBM	10.13	2.97	15.38	0.02
PCA-LightGBM	46.84	13.38	58.74	0.69
CNN-LightGBM	25.35	7.37	34.20	0.03
Bi-GRU-LightGBM	12.76	3.40	19.37	2.52
Proposed model	**9.63**	**2.64**	**14.27**	4.44

TABLE XI
SIGNIFICANCE TESTS ON THE TESTING SET

Models	*t*-Test	Friedman Test
Proposed model vs. SVR	0.000	
Proposed model vs. TCN	0.002	
Proposed model vs. LSTM	0.000	
Proposed model vs. Bi-GRU	0.000	p=0.000
Proposed model vs. LightGBM	0.000	(Reject H_0)
Proposed model vs. PCA-LightGBM	0.000	
Proposed model vs. CNN-LightGBM	0.000	
Proposed model vs. Bi-GRU-LightGBM	0.000	

H_0: There is no significant difference between the models.

training set is from 01-May-2010 to 17-Jul-2010, the validation set is from 18-Jul-2010 to 24-Jul-2010 and the testing set is from 25-Jul-2010 to 31-Jul-2010. Similarly, the parameters of each model are set as follows.

(1) TCN: The algorithm is built using the Keras library. The number of filters is 64, the size of the filter is 2 and the dilation factor is set to [1, 2, 4, 8].

(2) LightGBM: The number of trees is 790, the maximum depth is 3, the number of leaves is 8, the learning rate is set to 0.008, the bagging fraction is 0.12, the feature fraction is 0.98, the chosen model optimizer is Adam and the boosting model is the gbdt.

(3) CNN: The number of convolutional layers is 1, the number of filters in the convolutional layer is 8, the size of convolutional kernel is 2, the number of fully connected layers is 2 and the number of neurons in the fully connected layer is set to 32/1.

(4) LSTM: The number of hidden layers is 1 and the number of hidden nodes is set to 50.

(5) Bi-GRU: The number of hidden layers is 1 and the number of hidden nodes is set to 20.

(6) SVR: The kernel function is linear and the penalty factor is set to 1.

(7) PCA: The principal components are 3.

Similar to experiments 1 and 2, the statistical metrics and computational complexity of each model are computed and shown in Table XII. From these results, although the γ_{mae} and γ_{rmse} metrics have large values due to the high benchmark load of the coal mining industry, the proposed model still achieves the best forecasting performance. The statistical significance of the proposed model is presented in Table XIII. The significant tests are consistent with the aforementioned conditions. It can be concluded that there are significant differences between the proposed model and the contrast models.

TABLE XII
LOAD FORECASTING EVALUATION ON THE TESTING SET

Time	From 25-Jul-2010 to 31-Jul-2010			
Statistical metrics	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)	Computational time (s)
SVR	110.81	9.00	144.33	0.21
TCN	107.97	8.93	142.13	0.94
LSTM	110.74	9.18	144.98	0.02
Bi-GRU	109.54	9.02	143.98	0.04
LightGBM	107.66	8.87	142.04	0.01
PCA-LightGBM	152.95	12.95	191.51	1.25
CNN-LightGBM	110.87	9.13	144.42	0.04
Bi-GRU-LightGBM	107.41	8.83	142.94	3.49
Proposed model	**106.17**	**8.73**	**140.54**	0.76

TABLE XIII
SIGNIFICANCE TESTS ON THE TESTING SET

Models	*t*-Test	Friedman Test
Proposed model vs. SVR	0.000	
Proposed model vs. TCN	0.003	
Proposed model vs. LSTM	0.000	
Proposed model vs. Bi-GRU	0.000	p=0.000
Proposed model vs. LightGBM	0.000	(Reject H_0)
Proposed model vs. PCA-LightGBM	0.000	
Proposed model vs. CNN-LightGBM	0.000	
Proposed model vs. Bi-GRU-LightGBM	0.003	

H_0: There is no significant difference between the models.

E. Experiment 4: Load forecasting for the Irish industrial customer

In the public dataset, the smart meter data of an industrial customer collected from 01-Aug-2009 to 16-Sep-2009 are utilized to run simulations. Following the aforementioned splitting rules, the training set is from 01-Aug-2009 to 06-Sep-2009, the validation set is from 7-Sep-2009 to 11-Sep-2009 and the testing set is from 12-Sep-2009 to 16-Sep-2009. Similarly, we also adopt the aforementioned parameter tuning strategy to determine the appropriate parameters. The parameters of each model are listed as follows:

(1) TCN: The algorithm is built using the Keras library. The number of filters is 32, the size of the filter is 3 and the dilation factor is set to [1, 2, 4, 8].

(2) LightGBM: The number of trees is 120, the maximum depth is 3, the number of leaves is 7, the learning rate is set to 0.08, the bagging fraction is 0.6, the feature fraction is 0.96, the chosen model optimizer is Adam and the boosting model is the gbdt.

(3) CNN: The algorithm has a convolutional layer, a pooling layer and three full connected layers. The number of filters in the convolutional layer is 16, the size of convolutional kernel is 2 and the number of neurons in the fully connected layer is set to 128/16/1.

(4) LSTM: The number of hidden layers is 2 and the number of hidden nodes is set to 10/10.

(5) Bi-GRU: The number of hidden layers is 2 and the number of hidden nodes is set to 10/10.

(6) SVR: The kernel function is radial basis function (RBF), the penalty factor is set to 100 and the kernel parameter is set to 0.001.

(7) PCA: The principal components are 84.

To evaluate the effectiveness of the proposed method, the statistical metrics and computational complexity comparing with the contrast models are conducted. The evaluation results are presented in Table XIV. It can be observed that the proposed model outperforms the contrast models in the three statistical metrics. The computational time is also acceptable considering the proposed model has a significant improvement in forecasting accuracy. Table XV further presents the statistical significance of the proposed model. The *t*-test and Friedman test are implemented at the α=0.05 significance levels in one-tail conditions. It can be seen that the proposed model is significantly different from the contrast models.

TABLE XIV
LOAD FORECASTING EVALUATION ON THE TESTING SET

Time	From 12-Sep-2009 to 16-Sep-2009			
Statistical metrics	γ_{mae} (kW)	γ_{mape} (%)	γ_{rmse} (kW)	Computational time (s)
SVR	0.32	44.33	0.45	0.01
TCN	0.20	12.89	0.43	0.27
LSTM	0.23	15.02	0.48	0.10
Bi-GRU	0.21	13.78	0.43	0.20
LightGBM	0.20	12.34	0.45	0.01
PCA-LightGBM	0.35	35.15	0.61	0.10
CNN-LightGBM	0.22	15.60	0.45	0.10
Bi-GRU-LightGBM	0.20	12.64	0.44	6.31
Proposed model	**0.17**	**9.43**	**0.40**	2.61

TABLE XV
SIGNIFICANCE TESTS ON THE TESTING SET

Models	*t*-Test	Friedman Test
Proposed model vs. SVR	0.000	
Proposed model vs. TCN	0.000	
Proposed model vs. LSTM	0.000	
Proposed model vs. Bi-GRU	0.000	p=0.000
Proposed model vs. LightGBM	0.000	(Reject H_0)
Proposed model vs. PCA-LightGBM	0.000	
Proposed model vs. CNN-LightGBM	0.000	
Proposed model vs. Bi-GRU-LightGBM	0.000	

H_0: There is no significant difference between the models.

F. Discussion and analysis

In this paper, different types of industrial customers are used to run experimental simulations. The experimental results show that the proposed model outperforms the contrast models in terms of the forecasting accuracy. In addition, the following information can be obtained from the results:

(1) In the individual models, the LightGBM has a better effect on fitting the actual load and achieves the best prediction accuracy. Meanwhile, the computational time of the model is significantly less than those of other individual models. Theoretically, the reason is that the GOSS technology and histogram algorithm in the LightGBM model effectively reduce the data size and the number of features, which increases the forecasting efficiency. In addition, because the leaf-wise growth strategy with depth limitation plays an important role in reducing the training loss and overcoming overfitting, the forecasting accuracy of the LightGBM model is improved.

(2) In the hybrid models, the proposed model surpasses the Bi-GRU-LightGBM, CNN-LightGBM and PCA-LightGBM. It proves that the TCN is superior to the Bi-GRU, the CNN and PCA in feature extraction. Because of the dilated convolution, the TCN has a wider receptive field to capture the long-time-range historical data and temporal relationship. Meanwhile, the residual block is introduced to solve the training problem of the deep TCN model.

(3) The forecasting performance of the PCA-LightGBM and CNN-LightGBM models is worse than that of the individual LightGBM model. The reason is that the limitations of these feature extraction techniques lead to the loss of feature information in the load forecasting for industrial customers. The PCA model cannot extract the nonlinear features and the hidden temporal relationship in the features. Due to the limitation of the receptive field, the CNN model has a relatively poor ability to capture the long-time-range features. In addition, the CNN experiences challenges in the model training phase. Because the Bi-GRU model has the ability to capture bidirectional temporal information, the accuracy of the Bi-GRU-LightGBM model is improved. However, the extraction effect is still unsatisfactory due to the lack of convolution.

(4) The proposed model that integrates the superiority of the TCN and LightGBM achieves the best forecasting effect. The capabilities of the TCN in feature extraction and LightGBM in load forecasting are fully utilized. The proposed model yields improved forecast accuracy compared with the contrast models. The statistical significance tests prove that the proposed model is significantly different from the contrast models. Regarding the computational time, the proposed model requires more time due to additional feature extraction techniques, but the forecasting accuracy is improved. In practical applications, the forecasting accuracy should be mainly considered. Further, the predicted load profiles also illustrate that the proposed model can forecast occasional fluctuation by extracting the hidden feature information.

IV. Conclusion

This paper proposes a hybrid TCN-LightGBM model for short-term load forecasting for industrial customers. First, the electrical features are reconstructed with the actual variation and fluctuation trend by the fixed-length sliding time window model to make the networks look very far into the past to extract time-varying feature. Then, the electrical features are combined with the temperature and data factors as the input features of the TCN. Thus, the long-time-range historical time series and temporal relationship of features can be well extracted via the TCN. Finally, the forecasting advantages of the LightGBM model are fully utilized to improve the load forecasting accuracy of industrial customers. Multiple experimental results show that the proposed model has better robustness and a better forecasting effect compared with other contrast models.

In future work, the influence of different parameter tuning models on the load forecasting performance will be considered. A parameter optimization model with less time consumption and strong optimization ability will be selected to further improve the forecasting performance for different industrial customers. Besides, we will further consider the possibility of applying the proposed method to other forecasting scenarios (such as residential and commercial demand forecasting) and verify the forecasting effectiveness of the proposed method in other scenarios.

References

[1] National Bureau of Statistics of the People's Republic of China, *China energy statistical yearbook 2018*. Beijing, China: China Statistical Press, 2019, pp. 124-125.

[2] Y. Yang, M. Wang, and Y. Liu, "Peak-off-peak load shifting: Are public willing to accept the peak and off-peak time of use electricity price," *J. Clean. Prod.*, vol. 199, pp. 1066-1071, Oct. 2018.

[3] Y. Wang and Q. Chen, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2142-2151, May 2017.

[4] R. Jiao and T. Zhang, "Short-term non-residential load forecasting based on multiple sequences LSTM recurrent neural network," *IEEE Access*, vol. 6, pp. 59438-59448, 2018.

[5] J. W. Taylor, "Short-term load forecasting with exponentially weighted methods," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 458-464, Feb. 2012.

[6] J. F. Rendon-Sanchez and L. M. De Menezes, "Structural combination of seasonal exponential smoothing forecasts applied to load forecasting," *Eur J Oper Res.*, vol. 275, no. 3, pp. 916-924, Jun 2019.

[7] Kyung-Bin Song, Young-Sik Baek, and Dug Hun Hong, "Short-term load forecasting for the holidays using fuzzy linear regression method," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 96-101, Feb. 2005.

[8] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Trans. Smart Grid.*, vol. 5, no. 1, pp. 456-462, Jan. 2014.

[9] J. C. López, M. J. Rider, and Q. Wu, "Parsimonious short-term load forecasting for optimal operation planning of electrical distribution systems," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1427-1437, Mar. 2019.

[10] J. Zhang and Y. Wei, "Short term electricity load forecasting using a hybrid model," *Energy.*, vol. 158, pp. 774-781, Sep. 2018.

[11] M. Salami and F. Sobhani, "A hybrid short-term load forecasting model developed by factor and feature selection algorithms using improved grasshopper optimization algorithm and principal component analysis," *Electr Eng.*, vol. 102, no. 1, pp. 437-460, Mar. 2020.

[12] H. Jiang, Y. Zhang, and E. Muljadi, "A short-term and high-resolution distribution system load forecasting approach using support vector

regression with hybrid parameters optimization," *IEEE Trans. Smart Grid.*, vol. 9, no. 4, pp. 3341-3350, Jul. 2018.
[13] L. M. Sriram, M. Gilanifar, and Y. Zhou, "Causal markov elman network for load forecasting in multinetwork systems," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1434-1442, Feb. 2019.
[14] M. Raza, N. Mithulananthan, and J. Li, "Multivariate ensemble forecast framework for demand prediction of anomalous days," *IEEE Trans. Sustainable Energy.*, vol. 11, no. 1, pp. 27-36, Jan. 2020.
[15] W. Kong, Z. Y. Dong, and Y. Jia, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid.*, vol. 10, no. 1, pp. 841-851, Jan. 2019.
[16] X. Tang and Y. Dai, "Short-term power load forecasting based on multi-layer bidirectional recurrent neural network," *IET Gener. Transm. Distrib.*, vol. 13, no. 17, pp. 3847-3854, Sep. 2019.
[17] F. M. Bianchi, E. De Santis, and A. Rizzi, "Short-term electric load forecasting using echo state networks and PCA decomposition," *IEEE Access.*, vol. 3, pp. 1931-1943, Oct. 2015.
[18] Y. Lv, X. Xu, and R. Xu, "Research on short-term load forecasting approach for smart grid," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City. (ICITBS)*, Changsha, China, 2019, pp. 602-605.
[19] W. Sun and C. Zhang, "A hybrid BA-ELM model based on factor analysis and similar-day approach for short-term load forecasting," *Energies.*, vol. 11, no. 5, pp. 1282, May. 2018.
[20] J. Ospina, A. Newaz, and M. O. Faruque, "Forecasting of PV plant output using hybrid wavelet-based LSTM-DNN structure model," *IET. Renew. Power Gener.*, vol. 13, no. 7, pp. 1087-1095, May. 2019.
[21] K. Kang, H. Sun, and C. Zhang, "Short-term electrical load forecasting method based on stacked auto-encoding and GRU neural network," *Evol. Intel.*, vol. 12, pp. 385-394, Sep. 2019.
[22] K. Yan, X. Wang, and Y. Du, "Multi-step short-term power consumption forecasting with a hybrid deep learning strategy," *Energies.*, vol. 11, no. 11, pp. 3089, Nov. 2018.
[23] Imani and Maryam, "Long short-term memory network and support vector regression for electrical load forecasting," in *Proc. Int. Conf. Power Gener. Syst. Renew. Energy Technol. (PGSRET)*, Istanbul, Turkey, 2019.
[24] P. Lv and S. Liu, "EGA-STLF: A Hybrid short-term load forecasting model," *IEEE Access*, vol. 8, pp. 31742-31752, 2020.
[25] P. Tsinganos and B. Cornelis, "Improved gesture recognition based on sEMG signals and TCN," in *Proc. IEEE Int Conf Acoust Speech Signal Process Proc. (ICASSP)*, Brighton, United Kingdom, 2019, pp. 1169-1173.
[26] D. Joshua, B. Brett, and J. William. "Improving regional and teleseismic detection for single-trace waveforms using a deep temporal convolutional neural network trained with an array-beam catalog," *Sensors*, vol. 19, no. 3, Feb. 2019.
[27] Y. Cheng and Y. Liu, "HS-TCN: A semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in IoT," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, London, United Kingdom, 2019, pp. 1-7.
[28] P. Zhang, X. Wang, and J. Chen, "Spectral and temporal feature learning with two-stream neural networks for mental workload assessment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 6, pp. 1149-1159, Jun. 2019.
[29] J. Song and G. Xue, "Hourly heat load prediction model based on temporal convolutional neural network," *IEEE Access*, vol. 8, pp. 16726-16741, Jan. 2020.
[30] G. Ke and Q. Meng. "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Proces. Syst. (NIPS)*, Long Beach, CA, United states, 2017, pp. 3147-3155.
[31] J. Xu, W. Ding, and X. Hu, "VATE: A trade-off between memory and preserving time for high accurate cardinality estimation under sliding time window," *Comput. Commun.*, vol. 138, pp. 20-31, Apr. 2019.
[32] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, New Orleans, LA, United states, 2018, pp. 2159-2166.
[33] Z. Zhang, X. Wang, and C. Jung, "DCSR: Dilated convolutions for single image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1625-1635, Apr. 2019.
[34] K. Zhang, M. Sun, and T. Han, "Residual networks of residual networks: Multilevel residual networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1303-1314, Jun. 2018.
[35] K. Chen, K. Chen, and Q. Wang, "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid.*, vol. 10, no. 4, pp. 3943-3952, Jul. 2019.
[36] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Trans. Smart Grid.*, vol. 10, no. 2, pp. 2326-2329, Mar. 2019.
[37] X. Ma, J. Sha, and D. Wang, "Study on a prediction of P2P network loan default based on the machine learning lightGBM and XGboost algorithms according to different high dimensional data cleaning," *Electron Commer Res Appl.*, vol. 31, pp. 24-39, Aug. 2018.
[38] Y. Zhao, L. Ye, and P. Pinson, "Correlation-constrained and sparsity-controlled vector autoregressive model for spatio-temporal wind power forecasting," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5029-5040, Sep. 2018.
[39] C. Ye, Y. Ding, and P. Wang, "A data-driven bottom-up approach for spatial and temporal electric load forecasting," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1966-1979, May. 2019.
[40] Commission for Energy Regulation (CER). (2012). *CER Smart Metering Project Electricity Customer Behaviour Trial*. [Online]. Available: https://www.ucd.ie/issda
[41] J. Fiot and F. Dinuzzo, "Electricity demand forecasting by multi-task learning," *IEEE Trans. Smart Grid.*, vol. 9, no. 2, pp. 544-551, Mar. 2018.
[42] L. Wang and Z. Zhang, "Short-term electricity price forecasting with stacked denoising autoencoders," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2673-2681, Jul. 2017.
[43] G. Fan, L. Peng, and W. Hong, "Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model," *Appl. Energy.*, vol. 224, pp. 13-33, Aug. 2018.
[44] Z. Zhang, W. Hong, and J. Li, "Electric load forecasting by hybrid self-recurrent support vector regression model with variational mode decomposition and improved cuckoo search algorithm," *IEEE Access*, vol. 8, pp. 14642-14658, 2020.

Yuanyuan Wang (S'12-M'15) received the B.S. and M.Sc. degree in electrical engineering from Changsha University of Science and Technology, Changsha China, in 2004 and 2007 respectively, and the Ph.D. degree in electrical engineering from the College of Electrical Engineering, Guangxi University, Guangxi, China, in 2012.

Currently, she is an Associate Professor with the College of Electrical and Information Engineering, Changsha University of Science and Technology. Her research interests include machine learning, power system load forecasting, power system protection and control.

Jun Chen is currently working toward the M.S. degree in electrical engineering in the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China.

His research interests include deep learning and its application in load forecasting.

Xiaoqiao Chen is currently pursuing the PHD's degree in Department of Computing and Mathematical Science, California Institute of Technology, CA, USA.

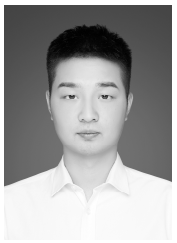
Her research interests include computer and applied mathematics, power system load forecasting.

Xiangjun Zeng (M'03) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, in 1993, the M.S. degree in electrical engineering from Wuhan University, Wuhan, China, in 1996, and the Ph.D. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, in 2001.

He was a Postdoctoral Fellow with Xuji Relay Company and Hong Kong Polytechnic University and a Visiting Professor with Nanyang Technological University, Singapore. He is currently a Professor and Dean of the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. His research focuses on real-time computer applications in power systems control and protection.

Yang Kong is currently working toward the M.S. degree in electrical engineering in the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China.

His current research interests include machine learning and its applications in load forecasting.



Shanfeng Sun is currently working toward the M.S. degree in electrical engineering in the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China.

His research interests include deep learning and its application in load forecasting.

Yongsheng Guo is currently pursuing the M.S. degree in electrical engineering in the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China.

His research interests machine learning and its applications in power systems, and data mining.

Ying Liu is currently pursuing the M.S. degree in electrical engineering in the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China.

Her research interests machine learning and its applications in power systems.